UNITED STATES DEPARTMENT OF COMMERCE
Patent and Trademark Office
Address: COMMISSIONER OF PATENTS AND TRADEMARKS
Washington, D.C. 20231

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.
09/226,939	01/08/99	VINCENT J	346872000500

EXAMINER	
LY,A	
ART UNIT	**PAPER NUMBER**
2172	10

DATE MAILED: 08/14/01

TM02/0814

MARTIN C. FLIESLER, ESQ.
FLIESLER, DUBB, MEYER & LOVEJOY LLP
FOUR EMBARCADERO CENTER
SUITE 400
SAN FRANCISCO CA 94111-4156

Please find below and/or attached an Office communication concerning this application or proceeding.

Commissioner of Patents and Trademarks

Office Action Summary	Application No. 09/226,939	Applicant(s) VINCENT ET AL.	
	Examiner Anh Ly	Art Unit 2172	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If the period for reply specified above is less than thirty (30) days, a reply within the statutory minimum of thirty (30) days will be considered timely.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133).
- Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *15 June 2001* .

2a)☐ This action is **FINAL**. 2b)☒ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-30* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☒ Claim(s) *1-8 and 30* is/are allowed.

6)☒ Claim(s) *9-29* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☐ The drawing(s) filed on _____ is/are: a)☐ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

11)☐ The proposed drawing correction filed on _____ is: a)☐ approved b)☐ disapproved by the Examiner.

If approved, corrected drawings are required in reply to this Office action.

12)☐ The oath or declaration is objected to by the Examiner.

Priority under 35 U.S.C. §§ 119 and 120

13)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____ .

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

14)☐ Acknowledgment is made of a claim for domestic priority under 35 U.S.C. § 119(e) (to a provisional application).

a) ☐ The translation of the foreign language provisional application has been received.

15)☐ Acknowledgment is made of a claim for domestic priority under 35 U.S.C. §§ 120 and/or 121.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO-1449) Paper No(s) _____ .
4) ☐ Interview Summary (PTO-413) Paper No(s). _____ .
5) ☐ Notice of Informal Patent Application (PTO-152)
6) ☐ Other:

DETAILED ACTION

Continued Prosecution Application

1. Receipt is acknowledged of the "conditional" request for a Continued Prosecution Application (CPA) filed on 06/15/2001 under 37 CFR 1.53(d) based on prior Application No. 09/226,939. Any "conditional" request for a CPA submitted as a separate paper is treated as an unconditional request for a CPA. Accordingly, the request for a CPA application is acceptable and a CPA has been established. An action on the CPA follows.

2. Claims 1-8 and 30 are allowable.

3. Claims 1-30 are pending in this application.

Claim Rejections - 35 USC § 103

The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

This application currently names joint inventors. In considering patentability of the claims under 35 U.S.C. 103(a), the examiner presumes that the subject matter of the various claims was commonly owned at the time any inventions covered therein were made absent any evidence to the contrary. Applicant is advised of the obligation under 37 CFR 1.56 to point out the inventor and invention dates of each claim that was

not commonly owned at the time a later invention was made in order for the examiner to consider the applicability of 35 U.S.C. 103© and potential 35 U.S.C. 102(f) or (g) prior art under 35 U.S.C. 103(a).

4. Claims 9 and 15 are rejected under 35 U.S.C. 103(a) as being unpatentable over US Patent No. 5,832,484 issued to Sankaran et al. (hereinafter as Sankaran) and in view of US Patent No. 5,970,490 issued to Morgenstern.

With respect to claim 9, Sankaran discloses dependency graph (a tree data structure) as claimed (see fig. 5B, 6A, 6B and 6C, col. 5, lines 1-12, col. 18, lines 4-16).

Sankaran does not explicitly indicate "querying a database catalog for the dependencies, and doing the query recursively until all basic dependencies are generated into a dependency tree."

However, Morgenstern discloses the querying for the database schema as claimed (col. 13, lines 39-45, and lines 53-57); recursion as claimed (col. 13, lines 18-24, and col. 14, lines 15-46).

Therefore, it would have been obvious to a person of ordinary skill in the art at the time the invention was made to combine the teachings of Sankaran with the teachings of Morgenstern so as to have a method for generating a basic dependency tree or graph because the combination would eliminate coding errors while declaring parameter data types in the anonymous blocks, at the same time initializing values for the declared parameters; allow for executing subprograms that take complex user defined types as parameters and also provide a complete execution facility that can be used to execute a code object in the database code object debugging environment.

With respect to claim 15, Sankaran discloses dependency graph (a tree data structure) as claimed (see fig. 5B, 6A, 6B and 6C, col. 5, lines 1-12, col. 18, lines 4-16).

Sankaran does not explicitly indicate, "to identify cyclic dependencies among database code objects."

However, Morgenstern discloses the cyclic directed graphs and dependency graph data structure as claimed (col. 20, lines 46-60).

Therefore, it would have been obvious to a person of ordinary skill in the art at the time the invention was made to combine the teachings of Sankaran with the teachings of Morgenstern so as to have a method for identifying cyclic dependencies among database code objects because the combination would eliminate coding errors while declaring parameter data types in the anonymous blocks, at the same time initializing values for the declared parameters; allow for executing subprograms that take complex user defined types as parameters and also provide a complete execution facility that can be used to execute a code object in the database code object debugging environment.

5. Claims 10-11, 13, and 16 are rejected under 35 U.S.C. 103(a) as being unpatentable over US Patent No. 5,832,484 issued to Sankaran et al. (hereinafter as Sankaran) and in view of US Patent No. 5,970,490 issued to Morgenstern, and further in view of US Patent No. 5,325,531 issued to McKeeman et al. (hereinafter as McKeeman).

With respect to claim 10, Sankaran in view of Morgenstern discloses a method of generating a basic dependency tree as discussed in claim 9.

Sankaran in view of Morgenstern does not explicitly indicate "part of a database code object debugging tool."

However, Mckeeman discloses a database code object debugging tool as claimed (col. 5, lines 19-56, and col. 6, lines, 1-12).

Therefore, it would have been obvious to a person of ordinary skill in the art at the time the invention was made to combine the teachings of Sankaran in view of Morgenstern with the teachings of McKeeman so as to have a method for debugging tool because the combination would eliminate coding errors while declaring parameter data types in the anonymous blocks, at the same time initializing values for the declared parameters; allow for executing subprograms that take complex user defined types as parameters and also provide a complete execution facility that can be used to execute a code object in the database code object debugging environment.

With respect to claim 11, Sankaran in view of Morgenstern discloses a method of generating a basic dependency tree as discussed in claim 9.

Sankaran in view of Morgenstern does not explicitly indicate, "to identify calling paths in a database."

However, Mckeeman discloses the paths in a database code coverage tool as claimed (col. 5, lines 19-56, and col. 6, lines, 1-12).

Therefore, it would have been obvious to a person of ordinary skill in the art at the time the invention was made to combine the teachings of Sankaran in view of Morgenstern with the teachings of McKeeman so as to have a method for identifying calling paths in a database because the combination would eliminate coding errors while declaring parameter data types in the anonymous blocks, at the same time initializing values for the declared parameters; allow for executing subprograms that take complex user defined types as parameters and also provide a complete execution facility that can be used to execute a code object in the database code object debugging environment.

With respect to claim 13, Sankaran in view of Morgenstern discloses a method of generating a basic dependency tree as discussed in claim 9.

Sankaran in view of Morgenstern does not explicitly indicate "a database code object testing tool."

However, Mckeeman discloses a testing tool as claimed (col. 2, lines 34-60, col. 5, lines 19-56, and col. 6, lines, 1-12).

Therefore, it would have been obvious to a person of ordinary skill in the art at the time the invention was made to combine the teachings of Sankaran in view of Morgenstern with the teachings of McKeeman so as to have a method for testing tool because the combination would eliminate coding errors while declaring parameter data types in the anonymous blocks, at the same time initializing values for the declared parameters; allow for executing subprograms that take complex user defined types as parameters and also provide a complete execution facility that can be used to execute a code object in the database code object debugging environment.

With respect to claim 16, Sankaran in view of Morgenstern discloses a method of generating a basic dependency tree as discussed in claim 9.

Sankaran in view of Morgenstern does not explicitly indicate "a dependency graph presentation tool."

However, Mckeeman discloses the tool as claimed (col. 2, lines 34-60, col. 5, lines 19-56, and col. 6, lines, 1-12).

Therefore, it would have been obvious to a person of ordinary skill in the art at the time the invention was made to combine the teachings of Sankaran in view of Morgenstern with the teachings of McKeeman so as to have a presentation tool because the combination would eliminate coding errors while declaring parameter data types in the anonymous blocks, at the same time initializing values for the declared parameters; allow for executing subprograms that take complex user defined types as

parameters and also provide a complete execution facility that can be used to execute a code object in the database code object debugging environment.

6. Claim 12 is rejected under 35 U.S.C. 103(a) as being unpatentable over US Patent No. 5,832,484 issued to Sankaran et al. (hereinafter as Sankaran) and in view of US Patent No. 5,970,490 issued to Morgenstern, and further in view of US Patent No. 5,692,193 issued to Jagannathan et al. (hereinafter as Jagannathan).

With respect to claim 12, Sankaran in view of Morgenstern discloses a method of generating a basic dependency tree as discussed in claim 9.

Sankaran in view of Morgenstern does not explicitly indicate "a database code object profiling tool."

However, Jagannathan discloses a database code object profiling tool as claimed (col. 11, lines 1-8).

Therefore, it would have been obvious to a person of ordinary skill in the art at the time the invention was made to combine the teachings of Sankaran in view of Morgenstern with the teachings of Jagannathan so as to have a method for profiling tool because the combination would eliminate coding errors while declaring parameter data types in the anonymous blocks, at the same time initializing values for the declared parameters; allow for executing subprograms that take complex user defined types as parameters and also provide a complete execution facility that can be used to execute a code object in the database code object debugging environment.

7. Claim 14 is rejected under 35 U.S.C. 103(a) as being unpatentable over US Patent No. 5,832,484 issued to Sankaran et al. (hereinafter as Sankaran) and in view of

US Patent No. 5,970,490 issued to Morgenstern, and further in view of US Patent No. 5,561,763 issued to Eto et al. (hereinafter as Eto).

With respect to claim 14, Sankaran in view of Morgenstern discloses a method of generating a basic dependency tree as discussed in claim 9.

Sankaran in view of Morgenstern does not explicitly indicate "to identify dependent objects that are INVALID in the database."

However, Eto discloses the INVALID in the database as claimed (col. 10, lines 12-21, col. 17, lines 4-57).

Therefore, it would have been obvious to a person of ordinary skill in the art at the time the invention was made to combine the teachings of Sankaran in view of Morgenstern with the teachings of Eto so as to have a method for identifying dependent objects that are INVALID in the database because the combination would eliminate coding errors while declaring parameter data types in the anonymous blocks, at the same time initializing values for the declared parameters; allow for executing subprograms that take complex user defined types as parameters and also provide a complete execution facility that can be used to execute a code object in the database code object debugging environment.

8. Claims 17-18, 20, 22-23, 25, 27-29 are rejected under 35 U.S.C. 103(a) as being unpatentable over US Patent No. 5,325,531 issued to McKeeman et al. (hereinafter as McKeeman) and in view of US Patent No. 5,926,819 issued to Doo et al. (hereinafter as Doo).

With respect to claim 17, McKeeman discloses recursion parser as claimed (col. 22, lines 62-67, and col. 23, lines 1-48); dependency graph as claimed (col. 5, lines 19-67, and col. 6, lines 1-12).

McKeeman does not explicitly indicate "database triggers, the dependency graph to identify DML statements that "fire" triggers so as to identify dependencies on triggers."

However, Doo discloses some databases implement replication by defining triggers when DML statements are executed as claimed (col. 1, lines 60-67, col. 2, lines 1-63, and col. 4, lines 45-65).

Therefore, it would have been obvious to a person of ordinary skill in the art at the time the invention was made to combine the teachings of McKeeman with the teachings of Doo so as to have a method of generating dependency information because the combination would eliminate coding errors while declaring parameter data types in the anonymous blocks, at the same time initializing values for the declared parameters; allow for executing subprograms that take complex user defined types as parameters and also provide a complete execution facility that can be used to execute a code object in the database code object debugging environment.

With respect to claim 18, Mckeeman discloses the paths in a database code coverage tool as claimed (col. 5, lines 19-56, and col. 6, lines, 1-12).

With respect to claim 20, Mckeeman discloses a testing tool as claimed (col. 2, lines 34-60, col. 5, lines 19-56, and col. 6, lines, 1-12).

With respect to claim 22, McKeeman discloses recursion parser as claimed (col. 22, lines 62-67, and col. 23, lines 1-48); dependency graph as claimed (col. 5, lines 19-67, and col. 6, lines 1-12).

McKeeman does not explicitly indicate, "to identify dependencies on triggers."

However, Doo discloses some databases implement replication by defining triggers when DML statements are executed as claimed (col. 1, lines 60-67, col. 2, lines 1-63, and col. 4, lines 45-65).

Therefore, it would have been obvious to a person of ordinary skill in the art at the time the invention was made to combine the teachings of McKeeman with the teachings of Doo so as to have a method of generating dependencies of code objects because the combination would eliminate coding errors while declaring parameter data types in the anonymous blocks, at the same time initializing values for the declared parameters; allow for executing subprograms that take complex user defined types as parameters and also provide a complete execution facility that can be used to execute a code object in the database code object debugging environment.

With respect to claim 23, Mckeeman discloses the paths in a database code coverage tool as claimed (col. 5, lines 19-56, and col. 6, lines, 1-12).

With respect to claim 25, Mckeeman discloses a testing tool as claimed (col. 2, lines 34-60, col. 5, lines 19-56, and col. 6, lines, 1-12).

With respect to claim 27, McKeeman discloses dependency graph (col. 17, lines 46-67, and col. 18, lines 1-4), and a code mechanism for generating a dependency

graph as claimed (col. 3, lines 34-67, col. 5, lines 19-67, col. 6, lines 1-12, col. 17, lines 20-67, and col. 18, lines 1-40).

McKeeman does not explicitly indicate "a digital computer, a database server coupled to the computer; a database coupled to the database server, the data including code object, specifications of packages, implementations of packages, specifications of type, implementations of types, and triggers..."

However, Doo discloses computer system item 100 (col. 4, lines 1-20), database server and database as claimed (col. 5, lines 43-67), implementations as claimed (col. 2, lines 18-25, and col. 7, lines 25-35), and triggers as claimed (col. Col. 4, lines 45-65).

Therefore, it would have been obvious to a person of ordinary skill in the art at the time the invention was made to combine the teachings of McKeeman with the teachings of Doo so as to have a method of generating dependency information because the combination would eliminate coding errors while declaring parameter data types in the anonymous blocks, at the same time initializing values for the declared parameters; allow for executing subprograms that take complex user defined types as parameters and also provide a complete execution facility that can be used to execute a code object in the database code object debugging environment.

With respect to claim 28, McKeeman discloses recursion parser as claimed (col. 22, lines 62-67, and col. 23, lines 1-48); dependency graph (col. 17, lines 46-67, and col. 18, lines 1-4), and a code mechanism for generating a dependency graph as claimed (col. 3, lines 34-67, col. 5, lines 19-67, col. 6, lines 1-12, col. 17, lines 20-67, and col. 18, lines 1-40).

McKeeman does not explicitly indicate "a digital computer, a database server coupled to the computer; a database coupled to the database server, the data including code object, specifications of packages, implementations of packages, specifications of type, implementations of types, and triggers..."

However, Doo discloses computer system item 100 (col. 4, lines 1-20), database server and database as claimed (col. 5, lines 43-67), implementations as claimed (col. 2, lines 18-25, and col. 7, lines 25-35), and triggers as claimed (col. Col. 4, lines 45-65).

Therefore, it would have been obvious to a person of ordinary skill in the art at the time the invention was made to combine the teachings of McKeeman with the teachings of Doo so as to have a method of generating dependency information because the combination would eliminate coding errors while declaring parameter data types in the anonymous blocks, at the same time initializing values for the declared parameters; allow for executing subprograms that take complex user defined types as parameters and also provide a complete execution facility that can be used to execute a code object in the database code object debugging environment.

With respect to claim 29, McKeeman discloses recursion parser as claimed (col. 22, lines 62-67, and col. 23, lines 1-48); dependency graph (col. 17, lines 46-67, and col. 18, lines 1-4), and a code mechanism for generating a dependency graph as claimed (col. 3, lines 34-67, col. 5, lines 19-67, col. 6, lines 1-12, col. 17, lines 20-67, and col. 18, lines 1-40).

McKeeman does not explicitly indicate "a digital computer, a database server coupled to the computer; a database coupled to the database server, the data including

code object, specifications of packages, implementations of packages, specifications of type, implementations of types, and triggers..."

However, Doo discloses computer system item 100 (col. 4, lines 1-20), database server and database as claimed (col. 5, lines 43-67), implementations as claimed (col. 2, lines 18-25, and col. 7, lines 25-35), and triggers as claimed (col. Col. 4, lines 45-65).

Therefore, it would have been obvious to a person of ordinary skill in the art at the time the invention was made to combine the teachings of McKeeman with the teachings of Doo so as to have a method of generating dependency information because the combination would eliminate coding errors while declaring parameter data types in the anonymous blocks, at the same time initializing values for the declared parameters; allow for executing subprograms that take complex user defined types as parameters and also provide a complete execution facility that can be used to execute a code object in the database code object debugging environment.

9. Claims 19 and 24 are rejected under 35 U.S.C. 103(a) as being unpatentable over US Patent No. 5,325,531 issued to McKeeman et al. (hereinafter as McKeeman) and in view of US Patent No. 5,926,819 issued to Doo et al. (hereinafter as Doo), and further in view of US Patent No. 5,692,193 issued to Jagannathan et al. (hereinafter as Jagannathan).

With respect to claim 19, Mckeeman in view of Doo discloses a method of generating dependency information as discussed in claim 17.

Mckeeman in view of Doo does not explicitly indicate "a database code object profiling tool."

However, Jagannathan discloses a database code object profiling tool as claimed (col. 11, lines 1-8).

Therefore, it would have been obvious to a person of ordinary skill in the art at the time the invention was made to combine the teachings of Mckeeman in view of Doo with the teachings of Jagannathan so as to have a method for profiling tool because the combination would eliminate coding errors while declaring parameter data types in the anonymous blocks, at the same time initializing values for the declared parameters; allow for executing subprograms that take complex user defined types as parameters and also provide a complete execution facility that can be used to execute a code object in the database code object debugging environment.

With respect to claim 24, Mckeeman in view of Doo discloses a method of generating dependency information as discussed in claim 22.

Mckeeman in view of Doo does not explicitly indicate "a database code object profiling tool."

However, Jagannathan discloses a database code object profiling tool as claimed (col. 11, lines 1-8).

Therefore, it would have been obvious to a person of ordinary skill in the art at the time the invention was made to combine the teachings of Mckeeman in view of Doo with the teachings of Jagannathan so as to have a method for profiling tool because the combination would eliminate coding errors while declaring parameter data types in the anonymous blocks, at the same time initializing values for the declared parameters; allow for executing subprograms that take complex user defined types as parameters and also provide a complete execution facility that can be used to execute a code object in the database code object debugging environment.

10. Claims 21 and 26 are rejected under 35 U.S.C. 103(a) as being unpatentable over US Patent No. 5,325,531 issued to McKeeman et al. (hereinafter as McKeeman) and in view of US Patent No. 5,926,819 issued to Doo et al. (hereinafter as Doo), and further in view of US Patent No. 5,561,763 issued to Eto et al. (hereinafter as Eto).

With respect to claim 21, Mckeeman in view of Doo discloses a method of generating dependency information as discussed in claim 17.

Mckeeman in view of Doo does not explicitly indicate "to identify dependent objects that are INVALID in the database."

However, Eto discloses the INVALID in the database as claimed (col. 10, lines 12-21, col. 17, lines 4-57).

Therefore, it would have been obvious to a person of ordinary skill in the art at the time the invention was made to combine the teachings of McKeeman in view of Doo with the teachings of Eto so as to have a method for identifying dependent objects that are INVALID in the database because the combination would eliminate coding errors while declaring parameter data types in the anonymous blocks, at the same time initializing values for the declared parameters; allow for executing subprograms that take complex user defined types as parameters and also provide a complete execution facility that can be used to execute a code object in the database code object debugging environment.

With respect to claim 26, Mckeeman in view of Doo discloses a method of generating dependency information as discussed in claim 22.

Mckeeman in view of Doo does not explicitly indicate "to identify dependent objects that are INVALID in the database."

However, Eto discloses the INVALID in the database as claimed (col. 10, lines 12-21, col. 17, lines 4-57).

Therefore, it would have been obvious to a person of ordinary skill in the art at the time the invention was made to combine the teachings of McKeeman in view of Doo with the teachings of Eto so as to have a method for identifying dependent objects that are INVALID in the database because the combination would eliminate coding errors while declaring parameter data types in the anonymous blocks, at the same time initializing values for the declared parameters; allow for executing subprograms that take complex user defined types as parameters and also provide a complete execution facility that can be used to execute a code object in the database code object debugging environment.

Contact Information

11. Any inquiry concerning this communication should be directed to Anh Ly whose telephone number is (703) 306-4527. The examiner can be reached on Monday - Friday from 8:00 AM to 4:00 PM.

If attempts to reach the examiner are unsuccessful, see the examiner's supervisor, Kim Vu, can be reached on (703) 305-4393.

Any response to this action should be mailed to:

Commissioner of Patents and Trademarks

Washington, D.C. 20231

or faxed to:

(703) 308-9051 (for formal communications intended for entry)

or:

(703) 305-9724 or (703) 308-6606 (for informal or draft communications, please label "PROPOSED" or "DRAFT")

Hand-delivered responses should be brought to Crystal Park II, 2121 Crystal Drive, Arlington, VA, Fourth Floor (receptionist).

Inquiries of a general nature or relating to the status of this application should be directed to the Group receptionist whose telephone number is (703) 305-9600.

AL/

August 8th, 2001

KIM VU
SUPERVISORY PATENT EXAMINER
TECHNOLOGY CENTER 2100